Beyond Knowledge Capture: Creating Useful Work-Centric Systems

Lynne P. Cooper
Jet Propulsion Laboratory
California Institute of Technology
4800 Oak Grove Drive, MS 302-231
Pasadena, CA 91109 USA
+1 818 393 3080
lynne.p.cooper@jpl.nasa.gov

Ann Majchrzak
School of Business Administration
University of Southern California
Los Angeles, CA 90089-1421 USA
+1 213 740 4023
majchrza@bus.usc.edu

ABSTRACT

Once you have successfully "captured" knowledge, the challenge then becomes one of creating an effective way to use that knowledge. Two high knowledge content systems developed at the Jet Propulsion Laboratory are presented as examples of work-centric systems, where the primary value to the user is in the content. Benefits are derived from designs that match the knowledge delivery mechanisms to the work processes the knowledge supports. These, and other high-value designs, can be differentiated based on whether the system operates in the user's foreground or background and whether the system is user- or system-paced.

Keywords

Knowledge management environments, work-centric systems, knowledge delivery

INTRODUCTION

The goal of knowledge capture efforts is to codify knowledge and make it available for use by a larger community. Often, this "captured knowledge" is used to populate a database, expert system, website, or other form of Knowledge Delivery System (KDS), which is then made available to a user community.

This paper describes a work-centric approach, which focuses on the knowledge itself and its use in a specific work context. The problem addressed is one of ensuring that captured knowledge is useful and provides value in the desired work context. Two systems currently being developed at the Jet Propulsion Laboratory, the Technical Questions Database and the Design Knowledge Capture Service, serve as examples of the work-centric approach and provide a basis for discussion of needed technology. The paper concludes with an analysis of the approach.

PROBLEM DESCRIPTION

There are many reasons why companies might be interested in a work-centric approach to knowledge-based systems (KBS). One such motivator may be the desire to ensure that knowledge is used. For example, Chevron's Knowledge Learning Officer reported that his organization's problem was not having knowledgebases; practically everyone has their own web site at Chevron. Instead, the problem was getting the knowledge used. Another motivator for a work-centric system might be the nature of the knowledge itself. Some knowledge may be sufficiently rich in content and context that simple knowledgebases are insufficient for sorting and identifying the appropriate knowledge. Finally, the motivator for moving toward a content-focused KBS might be that nature of the work process. Some work processes are inherently routine such that simple and stable categorization schemes are sufficient. Other work processes, such as those involving innovation, are less stable in their use of knowledge [9].

The mission of the Jet Propulsion Laboratory is exploration: "to do what no one has done before." The work is inherently innovative and technologically challenging. The push toward "better-faster-cheaper" projects has also led to the recognition that more-effective use of existing knowledge was critical to the success of future efforts. While standard knowledge management techniques for sharing best practices (e.g., document repositories, report templates) have been deployed to assist in the administration of projects, there is a recognized need for knowledge systems that support the creative work processes that are the lifeblood of the Laboratory, specifically, the design, development, and operation of space missions.

WORK-CENTRIC APPROACH

As implied in its name, the work-centric approach focuses primarily on the knowledge content and its value in a work process. As shown in Figure 1, there are five parts to the approach: Motivating Factors, Knowledge Capture, the

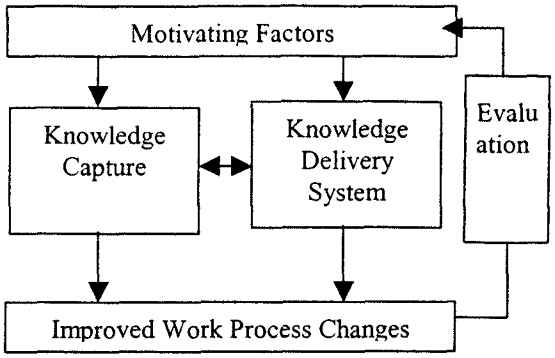


Figure 1. Work-centric Approach

Knowledge Delivery System, Improved Work Processes, and Evaluation.

Motivating factors are the events that lead to the realization that knowledge is needed to improve a work process. This motivation can be reactive (the need to prevent problems from re-occurring) or proactive (opportunity to create a competitive advantage). These factors drive what type of knowledge is needed and how it should be deployed.

Knowledge capture provides a means for one party to share its knowledge with another in a way that enables the knowledge to be applied in a meaningful way. To perform knowledge capture, the "knowledge" needs to be defined: What form does the knowledge take? How will it be organized, catalogued, indexed, or cross-referenced? What meta-knowledge is needed? The challenges associated with knowledge capture are well documented [10]. In a work-centric approach, the knowledge has intrinsic value, divorced from any system designed to present it.

The Knowledge Delivery System is responsible for storing and distributing the knowledge as required [8]. The combination of the knowledge and a system that makes it available to users should have a positive impact on work processes. It is important to understand how the process has changed, and to ensure that the changes that occur are the desired ones. To evaluate the system, one needs to assess its value to the intended users. The following two examples describe two work-centric systems currently in development at the Jet Propulsion Laboratory.

Examples

At the Jet Propulsion Laboratory (JPL), the majority of work falls into the category of "knowledge work" [1]. Capturing knowledge and making it available for broader application is important in meeting the goals of "faster-better-cheaper" and in ensuring that lessons learned on one project are incorporated on the next.

Technical Questions Database

The Technical Questions Database (TQ DB) is a web-accessible database containing sets of questions in technical areas employed at JPL to build and operate space missions. The questions are of the type expected at a peer review, and address detailed technical issues in domains such as thermal modeling, electronics design, materials properties, and deep space navigation. The knowledge consists of a set of questions in a given discipline, background information on why that question is important, a point of contact, and the organization responsible for that discipline. There are currently over 700 questions in over 70 technical disciplines.

The motivating factor for this work was the realization by upper management that there weren't sufficient domain experts available to support peer reviews, and therefore problems that should be caught at that level were being caught later in the development cycle when they were much more difficult to correct. While the database wasn't seen as a replacement for the experts, it was seen as a way of providing some of their insights to development teams.

The knowledge capture activities consisted of asking the functional organizations to submit questions, with the request framed as "What are the 10 (+/-) top questions that you would ask if you were at a peer review for work in your domain?"

The knowledge delivery system consists of a web front-end to the database that holds the questions. It enables users to browse, sort, search, view, select, and publish a report of questions in the represented technical disciplines. These are basic capabilities, and the primary expected mode of use is by browsing. In the event of a system failure, the entire set of questions is available in document format.

The TQ DB impacts the design and review processes. Users are expected to access the system during the course of their regular design activities to identify items they need to consider. In preparation for a review, the TQ DB serves as an "open book test" for the types of questions the designers can expect from their reviewers. It is also intended to help reviewers prepare by reminding them of the types of questions that are appropriate.

The TQ DB is an example of a system that operates as an advisor. The functionality is relatively simple, but use of the system "demands" the attention of the user to evaluate applicability and relevance, and to think about how to answer the questions relative to his/her design. The system is modeled on the peers and experts that attend reviews, therefore, the system approximates their behavior by "asking questions." In this sense, the knowledge representation is user natural, since this is the way the contributors (knowledge sources) and users would interact. The organization of the questions is also natural, based on the actual functional organization of the Laboratory. Users gain valuable information just from seeing which part of the organization contributed the questions.

The system has high perceived value because it meets the goal of helping to cover items that may otherwise be missed. The use of a browse (vs. search) paradigm requires users to identify categories that may be applicable to their domain, then review individual questions in those areas. This approach enables users to find questions in their areas that they *weren't* looking for -- the purpose of the resource.

The TQ DB is operational and available for use by JPL personnel. On-line mechanisms are being used to track usage and to solicit feedback on perceived usefulness, perceived ease of use, intention to use, and job relevance [3]. At the current size, the browse paradigm works well. However, when the size increases significantly, technology to support automated categorization, cross-referencing, and identification of closely-related items will become more important for both the users and maintainers of this resource.

Design Knowledge Capture Service

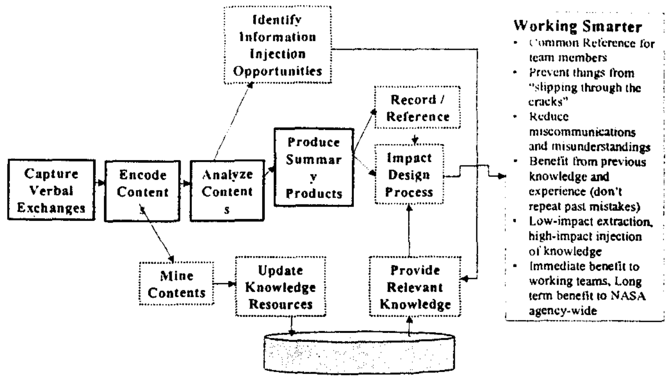
The Design Knowledge Capture Service (DeKCS) is a concept for capturing and processing the design knowledge expressed verbally during engineering design sessions. This knowledge includes options generated, the characteristics of the options, criteria used to evaluate the options, decisions made, rationale, and the impact of those decisions, as well as information that helps a design team operate more efficiently.

As illustrated in Figure 2, DeKCS is an ambitious concept. The primary path results in products of immediate benefit to the design team. The steps in this process are: (1) Transcribe verbal dialogue to written; (2) Encode the transcript; (3) Process the transcript; and (4) Create products. Ancillary activities include (1) identifying the need for external information and retrieving it; and (2) mining the transcript and products to support other activities (e.g., technology assessment).

The knowledge in DeKCS is two-fold: the design knowledge generated by the design team and the knowledge in the system that allows for the verbal dialogue to be processed.

The motivating factors for this work are the difficulties faced by design teams operating under the conditions of "better-faster-cheaper" which include smaller teams, tighter schedules, reduced resources, and increased performance expectations. This service is intended to (1) reduce the number of things that get accidentally overlooked; (2) capture and relate design options, decisions, and rationale; (3) reduce the need to "re-make" decisions; and (4) provide the means to introduce external knowledge to the team.

Figure 2: Design Knowledge Capture Service Concept



The knowledge delivery system is currently a manual process that uses some tools to assist in processing. The long-term vision is to create a system that can autonomously, in real-time, transcribe the design discussion, process it, and produce products (e.g., list of action items, list of open questions, decision map) during the session so they are available to the team during and after the session. The system will recognize when a question is asked that requires external input, and will formulate the query, analyze the responses, and volunteer the information to the team at an appropriate point. All this will be done in the background, without disrupting the flow of the design discussion. The performance of the knowledge delivery system will be assessed in terms of speed (realtime vs. after-the-fact), efficiency (manual vs. assisted vs. automated), and quality.

The work process will be changed because valuable information generated at the design sessions will be captured and made available for use immediately during the design session. Team members will be able to devote their whole attention to the design process rather than to note-taking. The record of the meeting will be more complete and accurate, reducing the problems that occur when individuals mentally filter events [4]. And the products produced will provide the means for synchronizing understanding. Since design work at JPL is inherently multi-disciplinary and involves complex decision-making, the DeKCS will help teams to work more efficiently and effectively.

In its ultimate incarnation, DeKCS will operate completely in the background until there is an explicit need to be met. Obviously, there is a gap between what today's technology can provide and this long-term vision. However, by using a combination of manual, assisted, near-real-time, and post processing, a baseline service can be deployed. As technology evolves, the framework will exist in which to incorporate it. The knowledge delivery system can be improved incrementally until it meets -- or exceeds -- the vision presented here.

The value of this system comes from the improvements to work processes. The system performs functions that each individual on the team currently does to varying degrees of effectiveness, and at a cost to the on-going discussion. The DeKCS approach shifts the cognitive load of "documenting" the design and "remembering" the rationale to a knowledge-based system, and provides easy-to-use artifacts that provide a group (vs. individual) perspective of the sessions.

The manual (low-tech) approach to implementing the service is currently being prototyped and tested. The initial goals are to better understand the types of products that will benefit the design teams, and to baseline the level of service (e.g., responsiveness, quality) needed for this to be perceived as valuable by users. At the same time, the technology needed to move toward the longer-term vision is being assessed. Each box in Figure 2 represents an opportunity for the use of new technology to move from post-processing to realtime, from manual to automated operations, and to improve the quality of the knowledge that is generated or injected.

A driver for long-term development of the DeKCS service is cost-effectiveness. As a manual process, the cost of using DeKCS service providers may exceed the value of the resulting products. So, while the service may be valuable during major design sessions, it is not feasible in the day-to-day design work that occurs. The system also relies on the participation of a skilled engineer to capture design information. This is not an easy slot to fill (most engineers would prefer to be doing the design rather than capturing it) so reducing the reliance on highly-skilled personnel is also critical for long-term feasibility. The strength of an evolvable system that begins as a primarily manual one is that system changes can be made to improve cost, and thereby increase the availability of the system to a wider range of users. Knowledge gained by the service providers during manual operations can be applied to improve the knowledge content and functionality of the system, and reduce dependency on highly-skilled personnel.

DISCUSSION

The work-centric approach to building knowledge-based systems firmly places the emphasis on the use of knowledge in a specific work context in order to improve that work process. The emphasis on understanding how knowledge will be used will sound familiar to those following user-centered, human factors-based, or cognitive engineering approaches. However, it is the matching of the knowledge, in combination with the system to deliver it, to the work process in which it is integrated that leads to systems that will be useful and valuable.

Each aspect of the approach presents its challenges. The following discussion places them in the context of previous work and identifies areas for future study.

Knowledge Capture

The challenges associated with knowledge capture are well documented [1]. In a work-centric approach, the knowledge has intrinsic value, divorced from any system designed to present it. Therefore, while practical concerns about how to represent the knowledge need to be considered, the most important aspect is capturing the knowledge in a way that captures and displays the content and context to facilitate appropriate reuse and recontextualization. One way to do this is to take a user personalization in which the knowledge can be displayed in a variety of different forms, corresponding to the application needs and decision making styles of different users [2]. For example, depending on the user, the same knowledge might be displayed using a picture, flow chart, hierarchical tree, semantic net, yellow pages of experts, set of heuristics, checklist of questions, or a story.

Another way is to capture the knowledge as closely as possible to its natural form. Rather than, for example, converting narratives to rules, leave them in their narrative form. Then provide users with the tools to synthesize, annotate, cross-reference, and otherwise layer their own cognitive map on top of the knowledge in its most natural form.

Knowledge mining technologies have tremendous potential for improving knowledge capture. In addition to processing documentation and artifacts, knowledge mining techniques could also be applied to the individual personalizations and cognitive maps to support knowledge discovery.

Knowledge Delivery System (KDS)

The Knowledge Delivery System often uses information technology for storing and distributing the knowledge. While this is common practice, there are other alternatives. For example, research librarians categorize and identify knowledge on demand. At Cap-Gemini, subject matter experts categorize, reconfigure, and notify others when consultant reports are added to the knowledge store. To do this work effectively, these experts and librarians must talk with the requestor and consultants to solicit the tacit knowledge required to optimize knowledge delivery. The value of having another person to talk to, who understands the knowledge content and can place it in context for the user, cannot be overrated.

In a work-centric system, the knowledge content is primary. The delivery system is secondary, and may even be best served by non-IT solutions. The delivery system adds incremental value, by making delivery faster, more efficient, or better, but take away the content and the remaining system is meaningless. As was illustrated in the DeKCS example, manual functions can, however, be overly expensive and susceptible to loss. The intent here is not to advocate one design over another. Rather, it is to call attention to alternatives not usually considered in IT-centric solutions. There is often greater value to the users in having a capability that intentionally evolves over time and relies more heavily on humans to perform knowledge-intensive functions, than in suffering through the growing pains of an incomplete technology.

Improved Work Processes

To improve work processes implies that the processes are known. This is a challenge since creative work is best described as an emergent work process. Markus, Majchrzak and Gasser [11] describe the characteristics of emergent knowledge work to include: the process unfolds unexpectedly over time, the people become involved in the process unexpectedly, and the knowledge used in the process can not be known in advance. With such a work process, it becomes a challenge to determine how best to design systems to match with an unpredictable work process. Markus et al. [11] suggest the adoption of a new set of design principles that allow the systems itself as well as its use to emerge over time, by, for example, integrating contradictory system requirements, representing relationships not rules, and componentizing the knowledge not just the architecture.

Evaluation

Evaluation is the process by which the usefulness and overall value of the system is determined. There are two key areas that need to be evaluated: (1) did the system address the motivating factors that inspired its creation? and (2) did the work processes actually change, and in a way that is considered an improvement? The question becomes one of what constitutes value in the given context.

Value is provided when the system ultimately facilitates the work process, which may be "off-line" and far-removed from system use [11]. For example, a new product may be created based on a conversation between two collocated individuals who have been informed by some well-designed knowledge base.

A system can also add value by creating a pleasurable experience for the user. Practitioners in usability engineering are moving toward concepts such as "joy of use" [5] and hedonic quality [7] to describe the need for systems to meet the affective needs of the users. A system that is fun -- or at least not annoying -- contributes to the overall value to the user. Additionally researchers have called for information system designers to strive for user delight rather than simply satisfaction [6] and user seduction [11] when users may not be initially inclined to believe that a system may be of value.

In addition to value, effort to use the system must be evaluated. Effort refers to the percentage of a user's total cognitive and attentional energy that she must devote to the user of the system. Our experience indicates that effort interacts as an inverted U-shaped curve with value, as shown in Figure 3. That is, people are willing to increase their effort in using a KBS as long as value is produced up to a certain maximum, at which point increased effort is seen to subtract from any additional value.

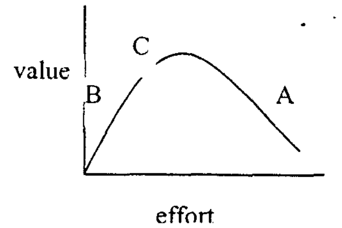


Figure 3: Relationship of Effort and Value

Given the relationship between effort and value, systems must first be designed to avoid the downward slope of effort (A in Figure 3). This can be done by following common human-computer interaction design principles, such as "no inputs without outputs", personalize interfaces to users' contexts and needs, filter information presented, never repeat, etc. [6][11][12]. Most search engines violate these basic principles by requiring users to repeatedly refine searches and wade through an overwhelming amount of information.

Having designed systems to avoid the downward slope of effort, systems must then be designed for the right break-even point on the upward slope of effort. Figure 3 shows two breakeven points: B & C. B is a low value, low effort system, such as a utility function for printing out search results. C is a high value, higher effort system such as the TQ DB. Either system might be fine, given the value that can be provided by the system to the work that people perform.

The systems we are most interested in are those that are of high value, based on the knowledge content of the system. These systems can be classified based on the level of conscious involvement required from the user (foreground vs. background) and who paces the system (user vs. system pacing). Figure 4 shows a 2x2 table arrayed on these variables and representative systems for each quadrant.

Two systems may to

Consciousness	User	System
Foreground	TQ DB	Training
Background	DeKCS	Models

Pacing

Figure 4. Classification of High-Value KBS

minimize effort, but one system may operate in the foreground while another may operate in the background. For example, a training simulation and the TQ DB both require the conscious involvement of the user. The training simulation, operating in the foreground, demands immediate attention from the user as the simulation proceeds. Similarly, the user of the TQ DB is responsible for evaluating material, consciously processing it to determine its relevance to the (for example) scientific instrument the user is developing.

In contrast, other systems need not command total attention and in fact operate in the background. At their best, systems working in the background understand the context in which the user is working, anticipate the needs of the user, and autonomously meet those needs. In the same way that a sou chef prepares the ingredients in the perfect amount, in the perfect form for use by the chef, or the way that an operating room nurse places the appropriate instrument into the hand of the surgeon as the request forms, a system working in the background provides the needed knowledge to the user in the needed format, at the needed time in anticipation of the need.

Why would a system be designed for the foreground versus the background? Our experience indicates that the decision is based on the degree to which the knowledge content is an active intervention to change behavior (in which case the system should be placed in the foreground) versus simply providing knowledge to reinforce and enhance work (in which case the system can be placed in the background). For example, a training simulation is placed in the foreground because it is intended to change behavior; a system that captures and anticipates knowledge needs of a surgeon in an operating room is placed in the background because it is intended to provide anticipated knowledge; not to change how the surgeon does her surgery.

Systems can also be differentiated based on what controls the operation of the system. Both the TQ DB and the DeKCS are user-paced. In the TQ DB, the users pull information from the system according to their needs. The users determine what information to look at next, whether to save an individual or set of questions to a report to review later, and can easily enter/exit the system at any time. The DeKCS is also user-paced, but in a different way. The user's work process (a design session) operates in the foreground while the system is responsible for keeping up with the process. The system is totally dependent on the work process proceeding. If the design session slows down or speeds up or becomes more intense, the system must respond accordingly.

Conversely, the system can be self-paced. The training simulation puts the user into a simulated environment where the user is responsible for responding to the stimuli created by the system. The training scenario drives the action and it is the user who must keep up with the process. Models and background simulations are also system-paced. These systems execute autonomously and use their knowledge content to reason about the environment and problems they were designed to solve. Monitoring systems, for example, are responsible for retrieving, processing, and reasoning about the inputs they receive. It's only in the event of an anomalous event that a user is alerted.

Understanding the strategic advantage of each option for a user's work process and motivational context is key to developing a system that will be evaluated well. Users with work processes that are largely unstructured (such as innovation) may benefit from a system in the foreground that "gets their attention" but they may bulk if it creates more structure than they are motivated to accept. The key is to match the components in Figure 1.

SUMMARY

Work-centric systems emphasize the primacy of the knowledge and place the system implementation firmly into a supporting role. This approach impacts the IT community because it requires a shift from technology to work process, and re-introduces "knowledge" as a driving factor in the design process. This paper discusses an approach to work-centric systems and highlights areas for special consideration (back/foreground, and user/system paced).

ACKNOWLEDGMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

1. Alavi, M. and Leidner, D.E. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. MIS Quarterly, 25 (1), 2001, 107-136.
2. Boland, R.J., Tenkasi, R.V. and Te'eni, D. Designing information technology to support distributed cognition. Organization Science, 1994, 5(3), 456-475.
3. Davis, F.D., Bagozzi, R.P., and Warshaw, P.R. User acceptance of computer technology: A comparison of two theoretical models. Management Science, 35(8), 1989, 982-1003.
4. Dougherty D. Interpretative barriers to successful product innovation in large firms. Organization Science. 1992; 3(2):179-202.
5. Glass, R. The looking glass. Available at http://www.sun.com.au/news/onsun/oct97/page6.html
6. Grudin, J. "Interactive systems: Bridging the gaps between developers and users. IEEE Computer, 24(4), 1991, 59-69.
7. Hassenzahl, M., Bue, A. and Burmester, M. Engineering Joy. IEEE Software, (January/February 2001), 70-76.
8. Leonard-Barton, D. Implementation as mutual adaptation of technology and organization. Research Policy, 17, 1988, 251-267.
9. Majchrzak, A, Rice, R.E., Malhotra, A., King, N., Ba, S. Technology adaptation: The case of a computer-supported inter-organizational virtual team. MIS Quarterly, 24(4), 2000, 569-600.

10. Markus, M.L. Toward a theory of knowledge reuse: Types of knowledge reuse situations and factors in reuse success. Journal of Management Information Systems, 18 (1), 2001, 57-93.

11. Markus, M.L., Majchrzak, A., and Gasser, L. A design theory for systems that support emergent knowledge processes. In A. Segars, J. Sampler, and R. Zmud (Eds.) Redefining the Organizational Roles of Information Technology. University of Minnesota Press, 2001.

12. Schneiderman, B. Codex, memex, genex: The pursuit of transformational technologies. International Journal of Human-Computer Interaction, 10(2), 1998, 87-106.